Combinatorial optimization

A Primer on the Basics

Bogdan Savchynskyy

03/07/2025

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Computer Vision and Learning Lab

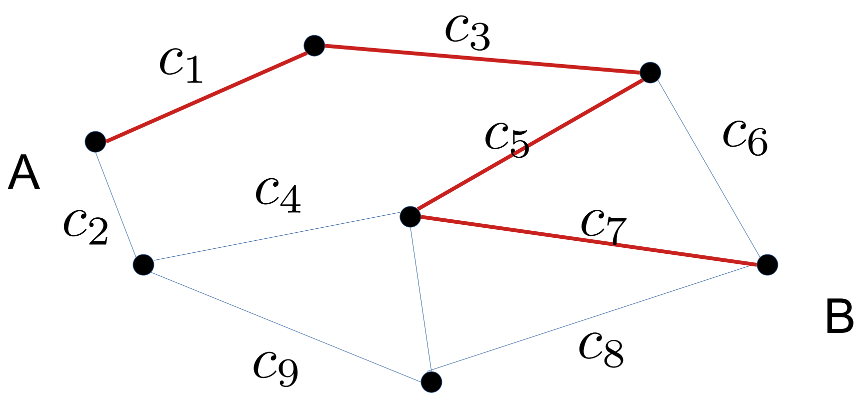
What is a combinatorial problem?

$\mathbb{Z}$ - integer numbers

Def.: Combinatorial problem: $\min_{\mathbf{z} \in S \subseteq \mathbb{Z}^n} f(\mathbf{z})$

What is a combinatorial problem?

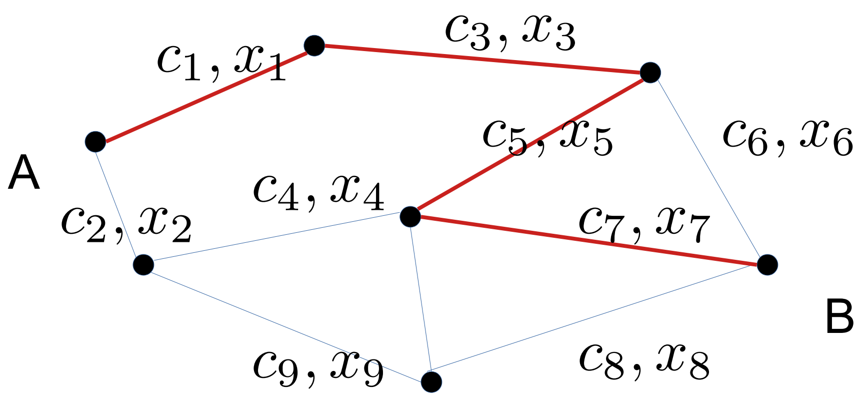
Given a weighted graph, find a **shortest path** between its two nodes



$c_i \in \mathbb{R}$

What is a combinatorial problem?

Given a weighted graph, find a **shortest path** between its two nodes

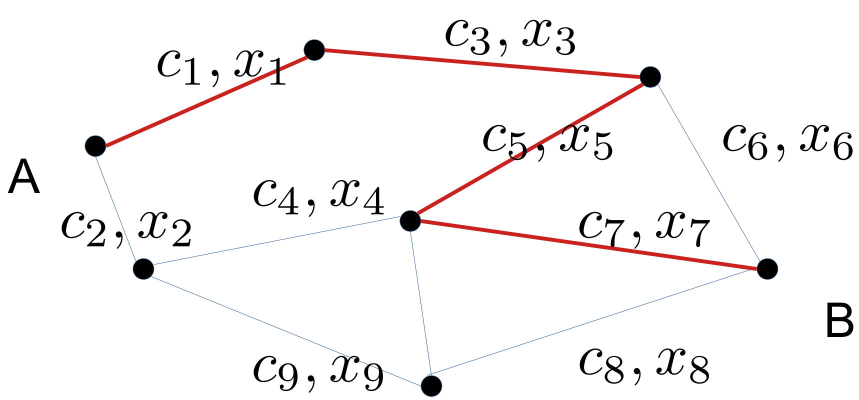


$$c_i \in \mathbb{R}$$

$$x_i \in \{0, 1\}$$

What is a combinatorial problem?

Given a weighted graph, find a **shortest path** between its two nodes



$$\min_{\mathbf{z} \in S \subseteq \mathbb{Z}^n} f(\mathbf{z})$$

$$c_i \in \mathbb{R}$$

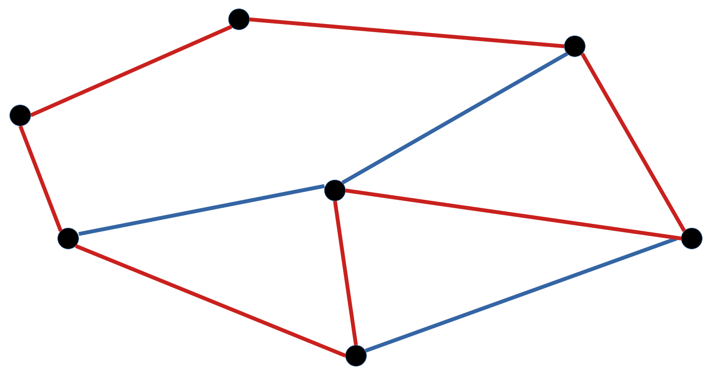
$$x_i \in \{0, 1\}$$

$$\mathbf{z} = (x_1, \ldots, x_n) \in \{0,1\}^n \subset \mathbb{Z}^n$$

$$f(\mathbf{z}) = \langle \mathbf{c}, \mathbf{z} \rangle$$

What is a combinatorial problem?

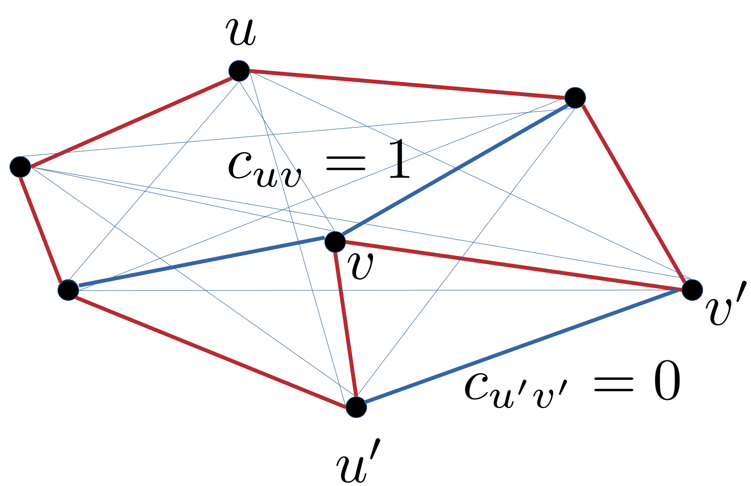
Hamiltonian cycle: Ist there a simple cycle containing all nodes of a given graph?



$\min_{\mathbf{z}\in S\subseteq\mathbb{Z}^n} \langle \mathbf{c}, \mathbf{z} \rangle = 0 \quad ?$

What is a combinatorial problem?

Hamiltonian cycle: Ist there a simple cycle containing all nodes of a given graph?



$$c_{uv} \in \{0, 1\}$$

$$x_{uv} \in \{0, 1\}$$

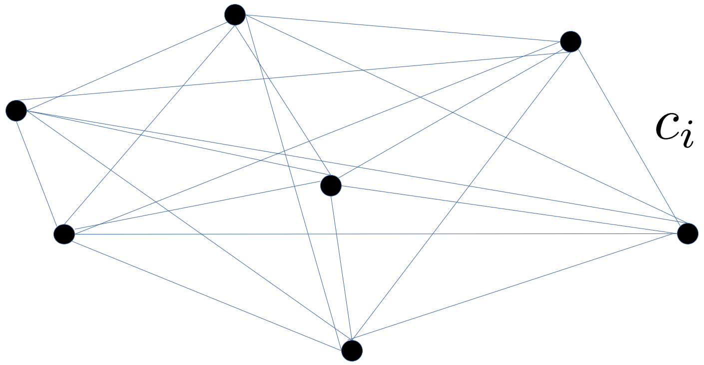
$$\{u, v\} \in \binom{\mathcal{V}}{2}$$

$$\mathbf{z} = (x_1, \ldots, x_n) \in \{0, 1\}^n \subset \mathbb{Z}^n$$

$$\min_{\mathbf{z}\in S\subseteq\mathbb{Z}^n} \langle \mathbf{c}, \mathbf{z} \rangle = 0 \quad ?$$

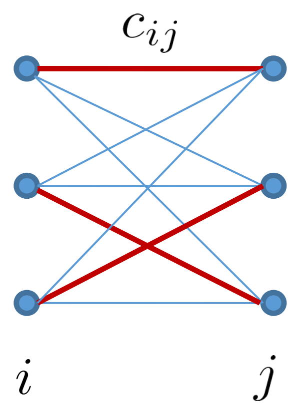
What is a combinatorial problem?

Traveling salesman: Find a shortest simple cycle containing all nodes in a fully connected graph.



What is a combinatorial problem?

Linear assignment (Weighted bipartite matching):



$$\min_{x \text{ is one-to-one matching}} \mathrm{c(x)}$$

What is a combinatorial problem?

Prime factorization:
Given an integer, represent it as a product of prime numbers

$$21 = 7 \cdot 3$$

What is a combinatorial problem?

0/1-Integer linear program:

$$\min_{\substack{\mathbf{x}\in\{0,1\}^n \\ A\mathbf{x}\leq\mathbf{b}}} \sum_i a_i x_i$$

What is a combinatorial problem?

Integer quadratic program:

$$\min_{\substack{\mathbf{x}\in\{0,1\}^n \\ A\mathbf{x}\leq\mathbf{b}}} \sum_i a_i x_i + \sum_{ij} c_i x_i x_j$$

What is a combinatorial problem?

Quadratic program: Not combinatorial

$$\min_{\substack{x \in [0,1]^n \\ A\mathbf{x} \leq \mathbf{b}}} \sum_i a_i x_i + \sum_{ij} c_i x_i x_j$$

Outline

- **Integer linear programming:**
  A universal language for combinatorial problems

- How off the shelf ILP solvers work

- What if off the shelf ILP solvers fail?

Universal representation for combinatorial problems

Theorem (Ibaraki'76): For $|S| < \infty$

$$\min_{\mathbf{z} \in S \subseteq \mathbb{Z}^n} f(\mathbf{z}) \quad \xrightarrow{\text{Representable as}} \quad \min_{\mathbf{x} \in \mathbb{Z}^n} \langle \mathbf{c}, \mathbf{x} \rangle$$

$$\text{s.t.: } A\mathbf{x} \leq \mathbf{b}$$

$$B\mathbf{x} = \mathbf{d}$$

Integer linear program (ILP)

Universal representation for combinatorial problems

Theorem (Ibaraki'76): For $|S| < \infty$

$$\min_{\mathbf{z} \in S \subseteq \mathbb{Z}^n} f(\mathbf{z})$$

Representable as

$$\min_{\mathbf{x} \in \mathbb{Z}^n} \langle \mathbf{c}, \mathbf{x} \rangle$$

$$\text{s.t.: } A\mathbf{x} \leq \mathbf{b}$$

$$B\mathbf{x} = \mathbf{d}$$

Integer linear program (ILP)

For |S|< ∞

$$\min_{\mathbf{x} \in \{0,1\}^n} \langle \mathbf{c}, \mathbf{x} \rangle$$

$$\text{s.t.: } A\mathbf{x} \leq \mathbf{b}$$

0/1 Integer linear program

Universal representation for combinatorial problems

Theorem (Ibaraki'76): For $|S| < \infty$

$$\min_{\mathbf{z} \in S \subseteq \mathbb{Z}^n} f(\mathbf{z})$$

Representable as

$$\min_{\mathbf{x} \in \mathbb{Z}^n} \langle \mathbf{c}, \mathbf{x} \rangle$$

$$\text{s.t.: } A\mathbf{x} \leq \mathbf{b}$$

$$B\mathbf{x} = \mathbf{d}$$

Integer linear program (ILP)

For |S|< ∞

$$\min_{\mathbf{x} \in \{0,1\}^n} \langle \mathbf{c}, \mathbf{x} \rangle$$

$$\text{s.t.: } A\mathbf{x} \leq \mathbf{b}$$

0/1 Integer linear program

$$\min_{\substack{\mathbf{x} \in \mathbb{Z}^n \\ \mathbf{y} \in \mathbb{R}^m}} \langle \mathbf{c}_x, \mathbf{x} \rangle + \langle \mathbf{c}_y, \mathbf{y} \rangle$$

$$\text{s.t.: } A\mathbf{x} + B\mathbf{y} \leq \mathbf{b}$$

Mixed Integer linear program (MILP)

Commercial and open-source MILP Solvers

GUROBI OPTIMIZATION

SCIP
Solving Constraint Integer Programs

IBM CPLEX

MOSEK

FICO® Xpress Optimization

nAG

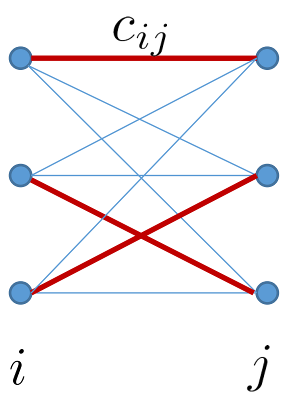
BARON Solver

hexaly

HiGHS

GLPK (GNU Linear Programming Kit)

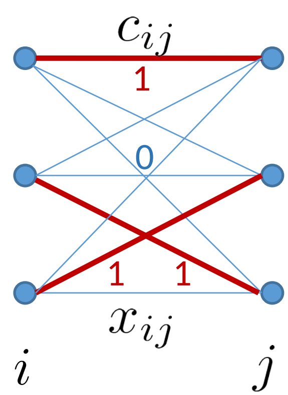
Examples: ILP formulation



linear assignment

c_{ij} - weights

Examples: ILP formulation



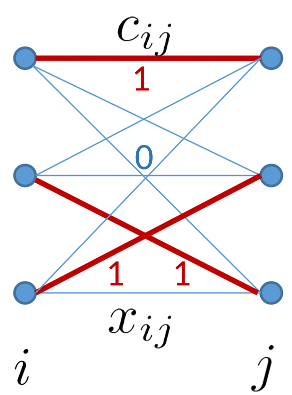
linear assignment

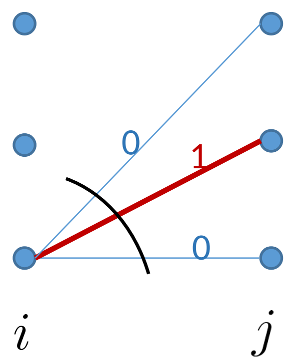
c_{ij} - weights

$x_{ij} \in \{0, 1\}$ - indicator variables

$\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$ - matching weight

Examples: ILP formulation





linear assignment

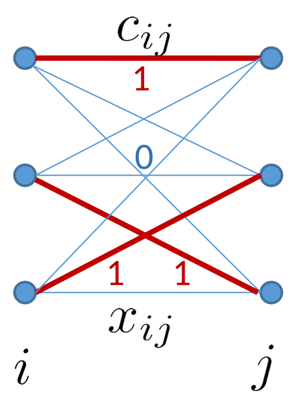
$$i: \sum_{j=1}^{n} x_{ij} = 1$$

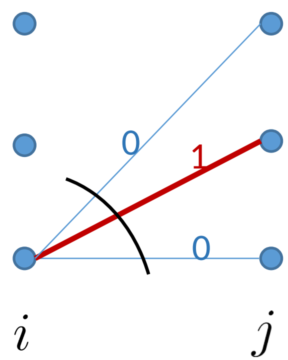
c_{ij} - weights

$x_{ij} \in \{0, 1\}$ - indicator variables

$\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$ - matching weight

Examples: ILP formulation





$$i : \sum_{j=1}^{n} x_{ij} = 1$$

$$\min_{x \in \{0,1\}^{n \times n}} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\text{s.t.:} \sum_{j=1}^{n} x_{ij} = 1 \ \forall i$$

$$\sum_{i=1}^{n} x_{ij} = 1 \ \forall j$$

linear assignment

c_{ij} - weights

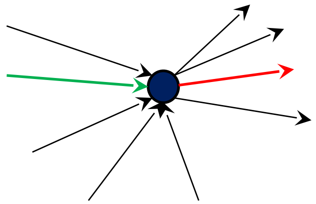
$x_{ij} \in \{0, 1\}$ - indicator variables

$\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$ - matching weight

Examples: ILP formulation

$$\sum_i x_i \leq 1 \quad \text{- at most one}$$

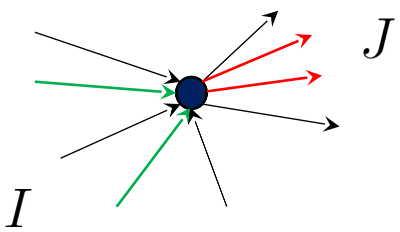
$$\sum_i x_i \geq 1 \quad \text{- at least one}$$



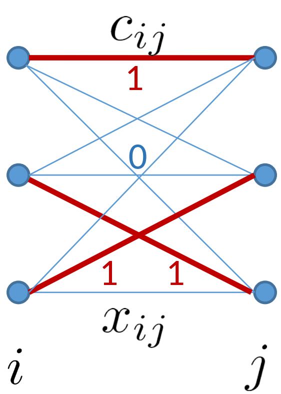
$$\sum_i x_i = 1 \quad \text{- exactly one}$$

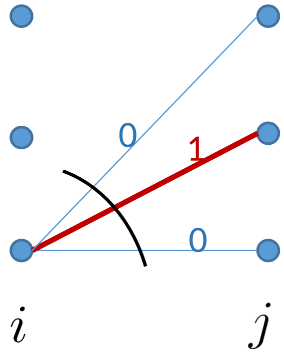
$$\sum_i x_i = k \quad \text{- exactly } k$$

$$\sum_{i \in I} x_i = \sum_{j \in J} x_j \quad \text{- flow conservation}$$



Example: Linearization trick



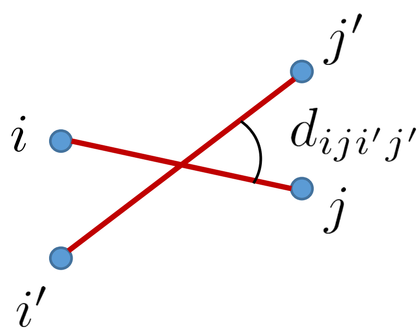


$$i: \sum_{j=1}^{n} x_{ij} = 1$$

$$\min_{x \in \{0,1\}^{n \times n}} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{iji'j'} d_{iji'j'} x_{ij} x_{i'j'}$$

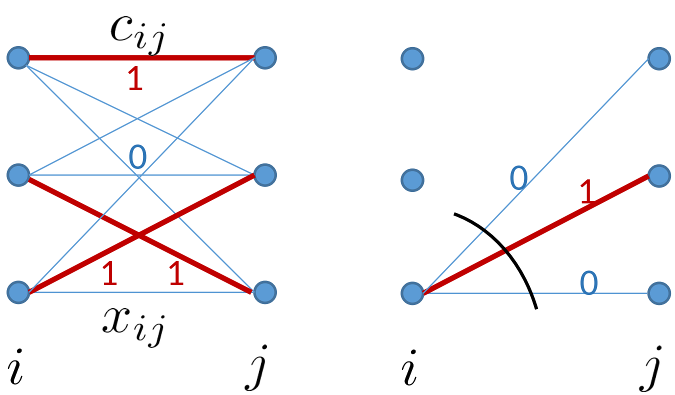
$$\text{s.t.:} \sum_{j=1}^{n} x_{ij} = 1 \ \forall i$$

$$\sum_{i=1}^{n} x_{ij} = 1 \ \forall j$$



Quadratic assignment

Example: Linearization trick

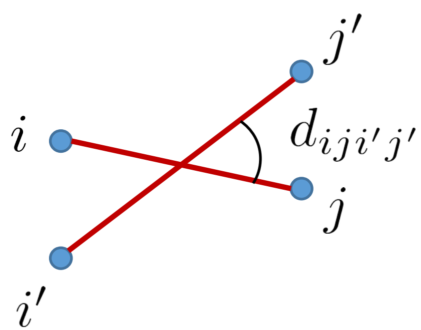


$$\min_{x\in\{0,1\}^{n\times n}} \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{iji'j'} d_{iji'j'}x_{ij}x_{i'j'}$$

$$\text{s.t.:} \sum_{j=1}^{n} x_{ij} = 1 \ \forall i$$

$$\sum_{i=1}^{n} x_{ij} = 1 \ \forall j$$

$$i: \sum_{j=1}^{n} x_{ij} = 1$$

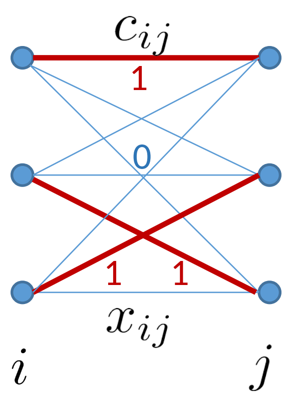


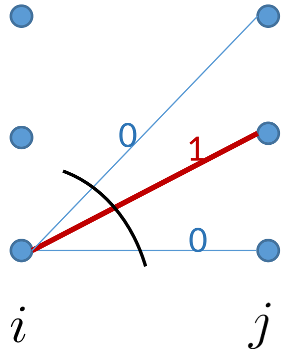
$$y_{iji'j'} := x_{ij}x_{i'j'}$$

Non-linear constraint!

Quadratic assignment

Example: Linearization trick



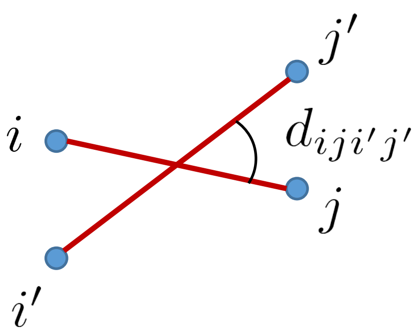


$$\min_{x\in\{0,1\}^{n\times n}} \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{iji'j'} d_{iji'j'}x_{ij}x_{i'j'}$$

$$\text{s.t.:} \sum_{j=1}^{n} x_{ij} = 1 \ \forall i$$

$$\sum_{i=1}^{n} x_{ij} = 1 \ \forall j$$

$$i: \ \sum_{j=1}^{n} x_{ij} = 1$$



$$y_{iji'j'} := x_{ij}x_{i'j'}$$

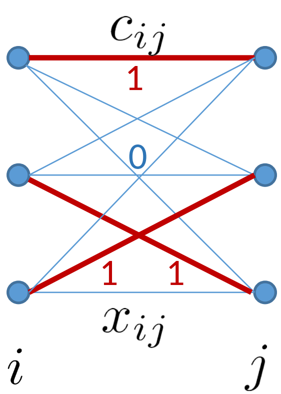
Non-linear constraint!

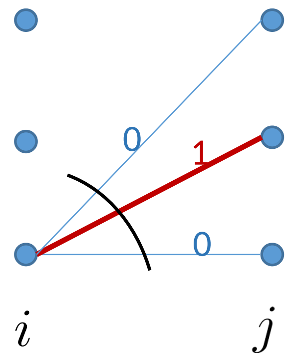
$$y_{iji'j'} \le x_{ij}$$

$$y_{iji'j'} \le x_{i'j'}$$

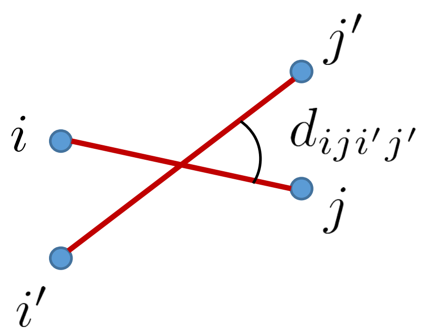
$$y_{iji'j'} \ge x_{ij} + x_{i'j'} - 1$$

Quadratic assignment





$$i: \quad \sum_{j=1}^{n} x_{ij} = 1$$



Quadratic assignment

$$\min_{x \in \{0,1\}^{n \times n}} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{iji'j'} d_{iji'j'} y_{iji'j'}$$

$$\text{s.t.:} \sum_{j=1}^{n} x_{ij} = 1 \ \forall i$$

$$\sum_{i=1}^{n} x_{ij} = 1 \ \forall j$$

$$\forall i, j, i', j':$$

$$y_{iji'j'} \leq x_{ij}$$

$$y_{iji'j'} \leq x_{i'j'}$$

$$y_{iji'j'} \geq x_{ij} + x_{i'j'} - 1$$

There are better linearizations...

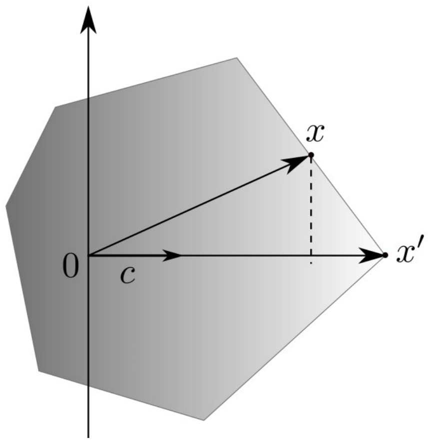
Outline

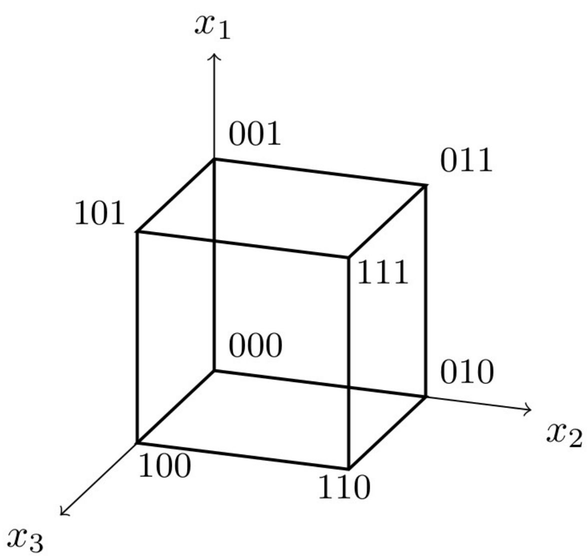
- Integer linear programming:
  A universal language for combinatorial problems
- **How off the shelf ILP solvers work**
- What if off the shelf ILP solvers fail?

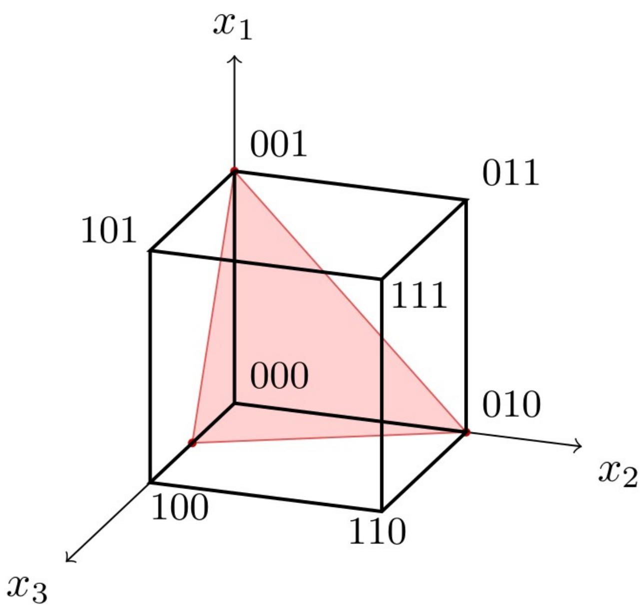
Geometry of ILP

$$\max_{\mathbf{x}\in\{0,1\}^n} \langle \mathbf{c}, \mathbf{x} \rangle$$

$$\text{s.t.: } A\mathbf{x} \leq \mathbf{b}$$







$A\mathbf{x} \leq \mathbf{b}$ $\{0,1\}^n$ $\{\mathbf{x} \in \{0,1\}^n : A\mathbf{x} \leq \mathbf{b}\}$

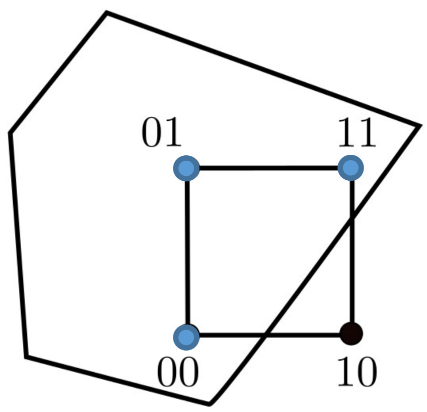
Solution = vertex

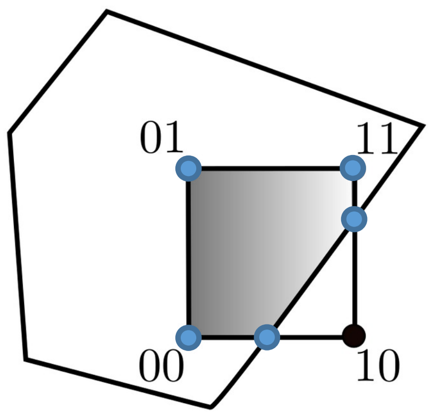
How MILP solvers work:Linear program (LP) relaxation

$$\max_{\mathbf{x}\in\{0,1\}^n} \langle \mathbf{c}, \mathbf{x} \rangle \quad \text{s.t.: } A\mathbf{x} \le \mathbf{b} \qquad \le \qquad \max_{\mathbf{x}\in[0,1]^n} \langle \mathbf{c}, \mathbf{x} \rangle \quad \text{s.t.: } A\mathbf{x} \le \mathbf{b}$$

Integer linear program

~~Integer~~ linear program (LP) relaxation





solution = 0/1 vertex

0/1 solution = vertex

Systematic search: Branch-and-Bound

$$\max_{\mathbf{x}\in\{0,1\}^7} \quad (40,\ 60,\ 10,\ 10,\ 3,\ 20,\ 60)\cdot\mathbf{x}$$
$$\text{s.t. } 100 \geq \quad (40,\ 50,\ 30,\ 10,\ 10,\ 40,\ 30)\cdot\mathbf{x}$$

$f^{int} := -\infty$

$\mathbf{x} = (\frac{1}{2}, 1, 0, 0, 0, 0, 1)$
Obj. val. $= 140$

Solve LP relaxation: f^{LP}

- $f^{LP} < f^{int}$?

  No

- $x^{LP} \in \{0,1\}^n$?

  No

- Branch

Systematic search: Branch-and-Bound

$$\max_{\mathbf{x}\in\{0,1\}^7} \; (40,\; 60,\; 10,\; 10,\; 3,\; 20,\; 60)\cdot\mathbf{x}$$
$$\text{s.t. } 100 \geq (40,\; 50,\; 30,\; 10,\; 10,\; 40,\; 30)\cdot\mathbf{x}$$

$f^{int} := -\infty$

$\mathbf{x} = (\frac{1}{2}, 1, 0, 0, 0, 0, 1)$

Obj. val. $= 140$

$x_1 = 0$: $(0, 1, 0, 1, 0, \frac{1}{4}, 1)$, 135

$x_1 = 1$: $(1, \frac{3}{5}, 0, 0, 0, 0, 1)$, 136

Solve LP relaxation: f^{LP}

- $f^{LP} < f^{int}$?

  No

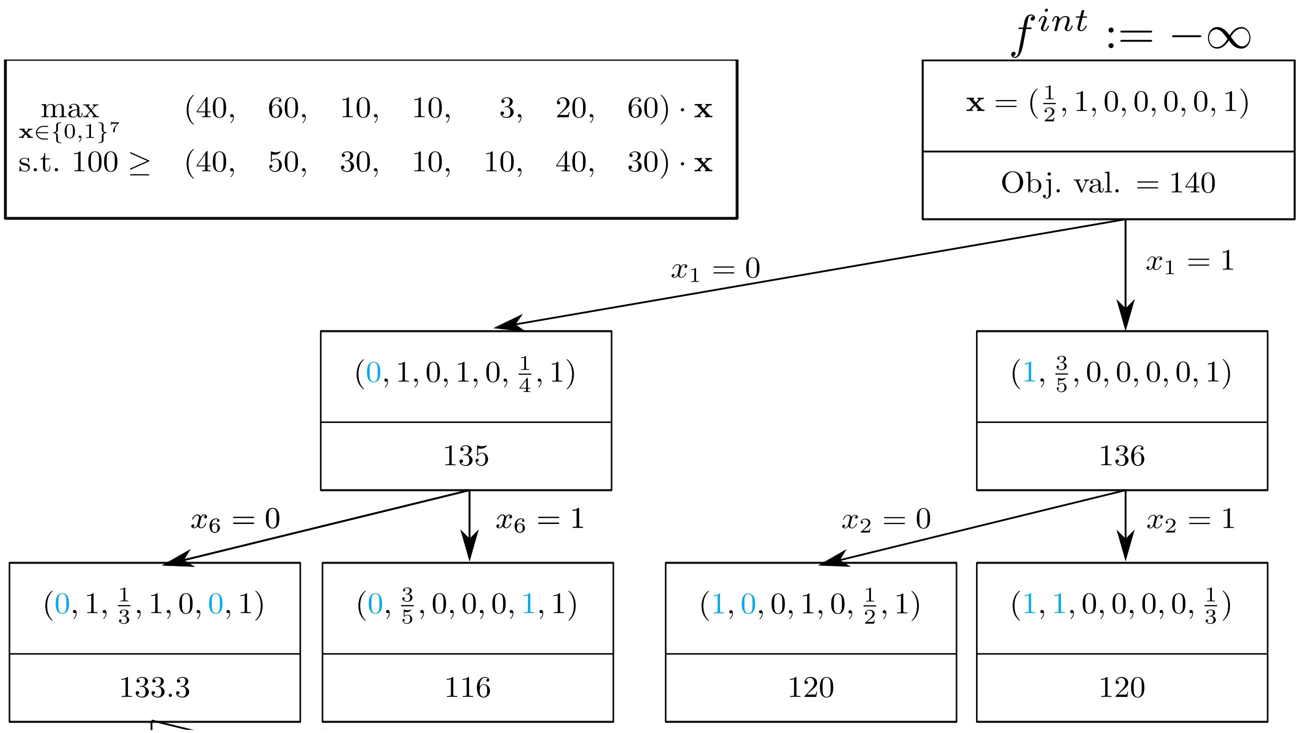
- $x^{LP} \in \{0,1\}^n$?

  No

- Branch

Systematic search: Branch-and-Bound

$$\max_{\mathbf{x}\in\{0,1\}^7} \quad (40,\ 60,\ 10,\ 10,\ 3,\ 20,\ 60)\cdot\mathbf{x}$$
$$\text{s.t. } 100 \geq \quad (40,\ 50,\ 30,\ 10,\ 10,\ 40,\ 30)\cdot\mathbf{x}$$

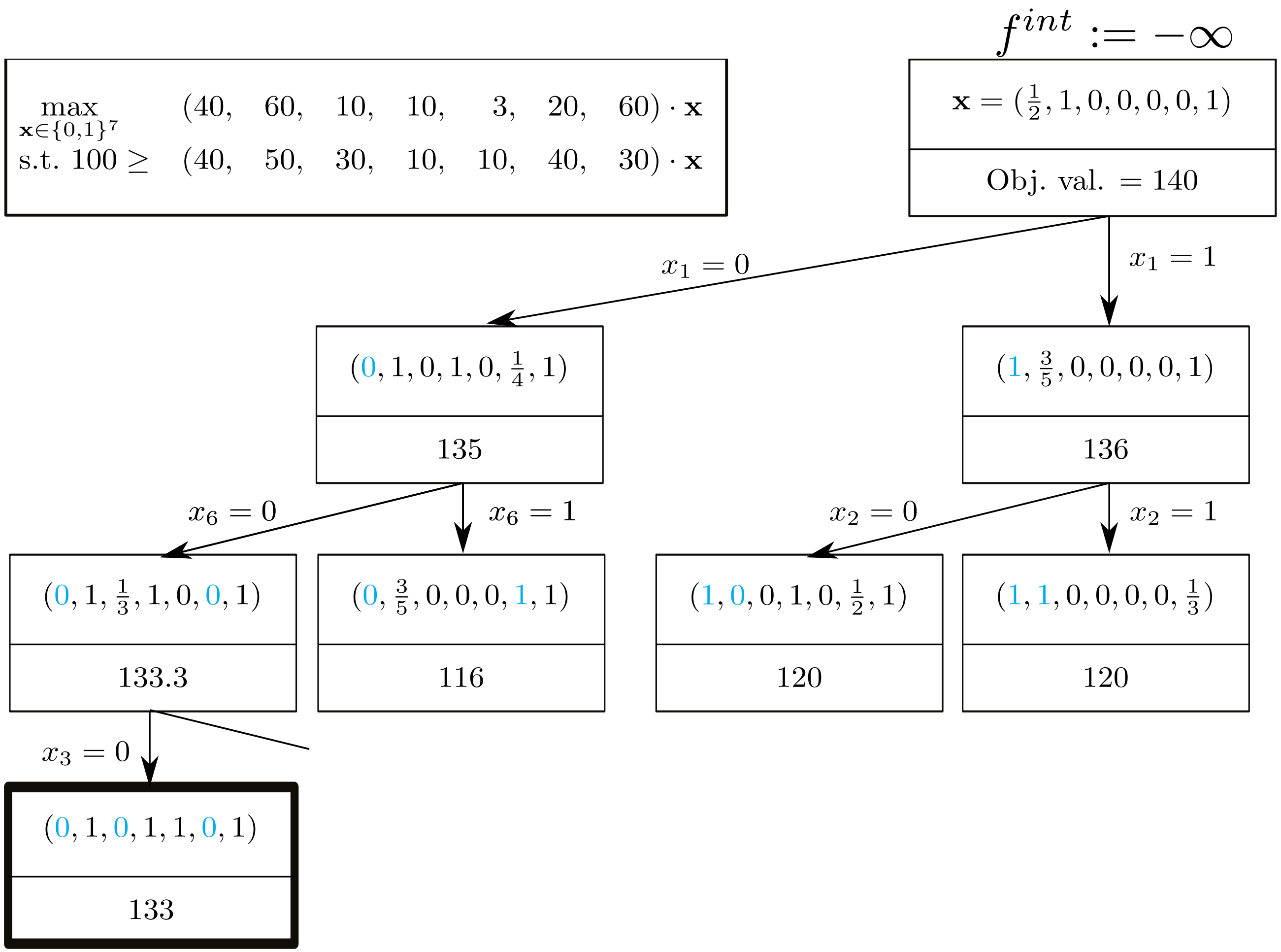


Solve LP relaxation: f^{LP}

- $f^{LP} < f^{int}$?
  No
- $x^{LP} \in \{0,1\}^n$?
  No
- Branch

Systematic search: Branch-and-Bound

$$\max_{\mathbf{x}\in\{0,1\}^7} \ (40,\ 60,\ 10,\ 10,\ 3,\ 20,\ 60)\cdot\mathbf{x}$$
$$\text{s.t. } 100 \geq (40,\ 50,\ 30,\ 10,\ 10,\ 40,\ 30)\cdot\mathbf{x}$$



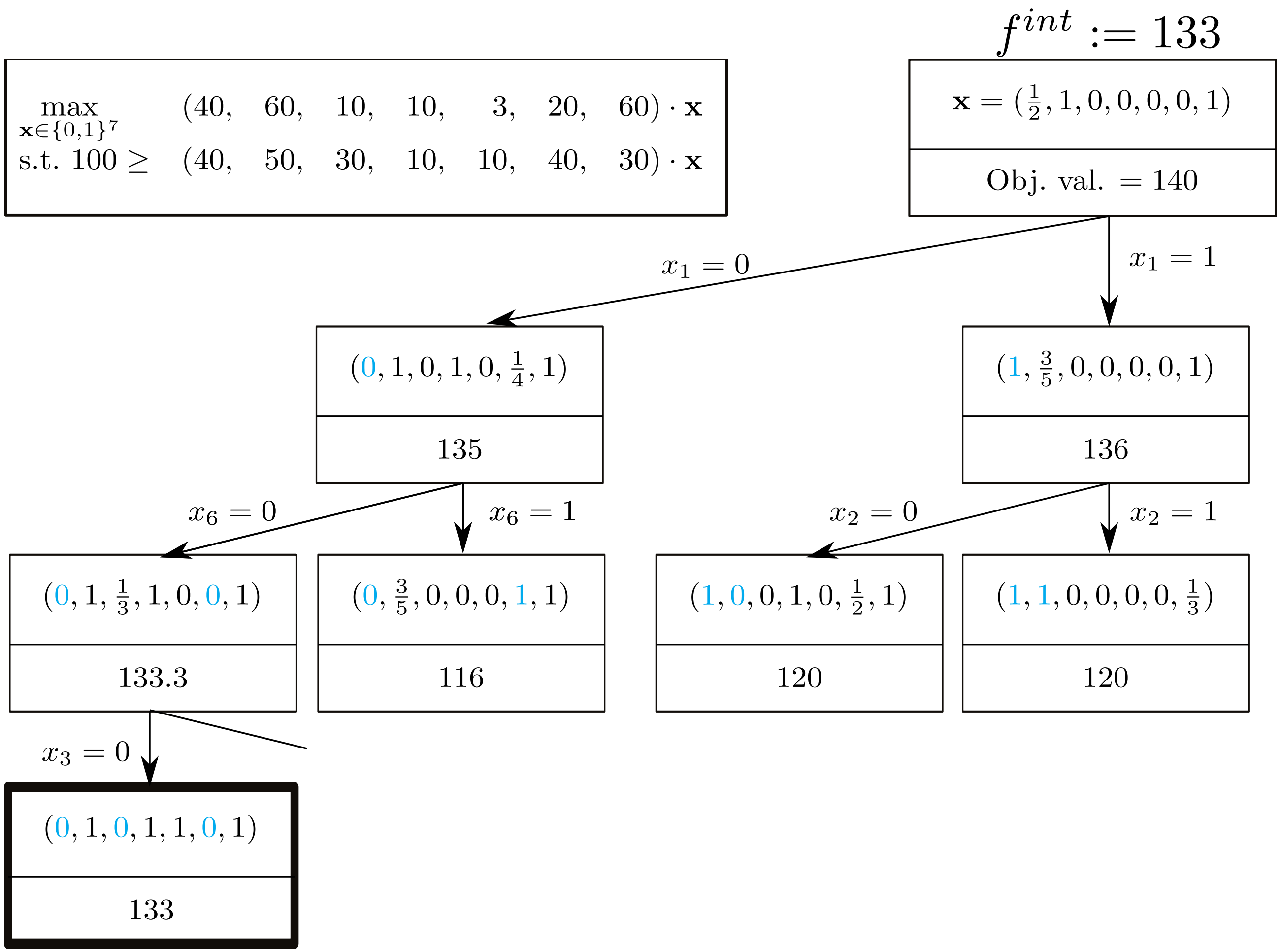
Solve LP relaxation: f^{LP}

- $f^{LP} < f^{int}$?

  No

Systematic search: Branch-and-Bound

$$\max_{\mathbf{x}\in\{0,1\}^7} \ (40,\ 60,\ 10,\ 10,\ 3,\ 20,\ 60)\cdot\mathbf{x}$$
$$\text{s.t. } 100 \geq (40,\ 50,\ 30,\ 10,\ 10,\ 40,\ 30)\cdot\mathbf{x}$$



Solve LP relaxation: f^{LP}

- $f^{LP} < f^{int}$?

  No

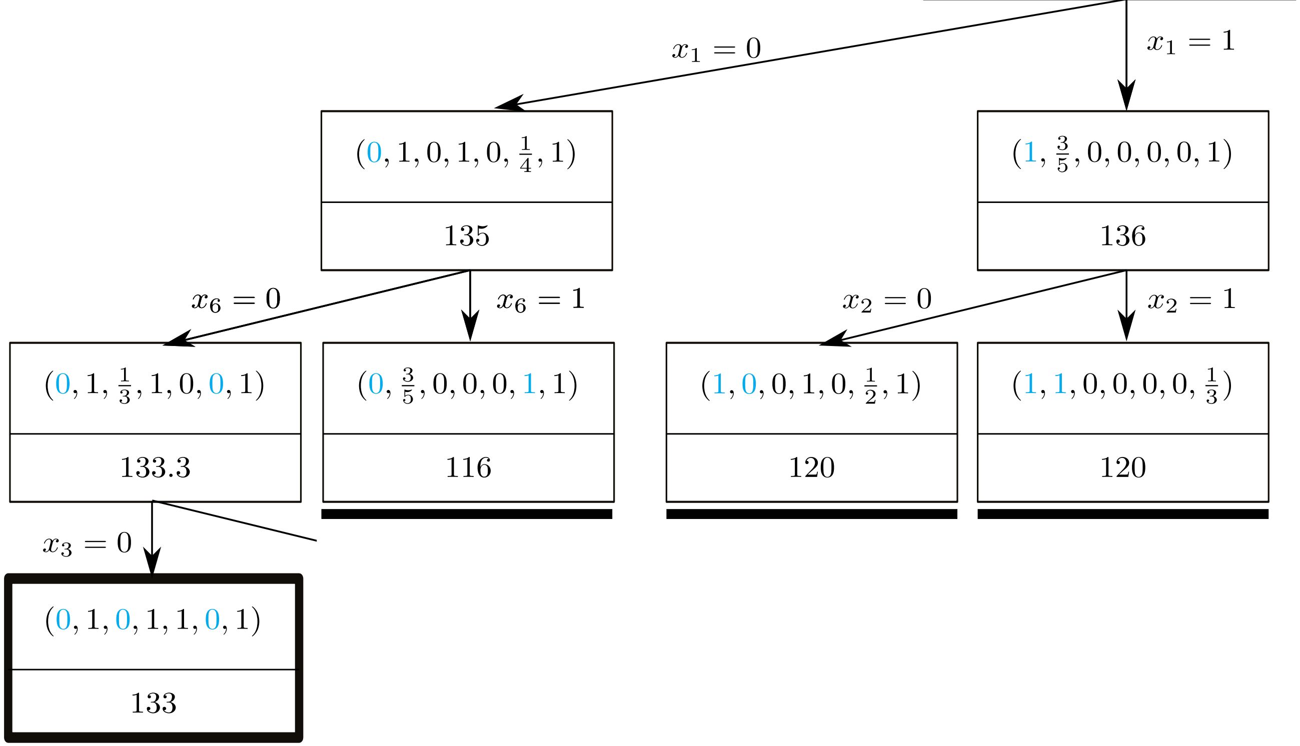
- $x^{LP} \in \{0,1\}^n$?

  Yes: $f^{int} := f^{LP}$

  terminate branch

Systematic search: Branch-and-Bound

$$\max_{\mathbf{x}\in\{0,1\}^7} \quad (40,\ 60,\ 10,\ 10,\ 3,\ 20,\ 60)\cdot\mathbf{x}$$
$$\text{s.t. } 100 \geq (40,\ 50,\ 30,\ 10,\ 10,\ 40,\ 30)\cdot\mathbf{x}$$



Solve LP relaxation: f^{LP}

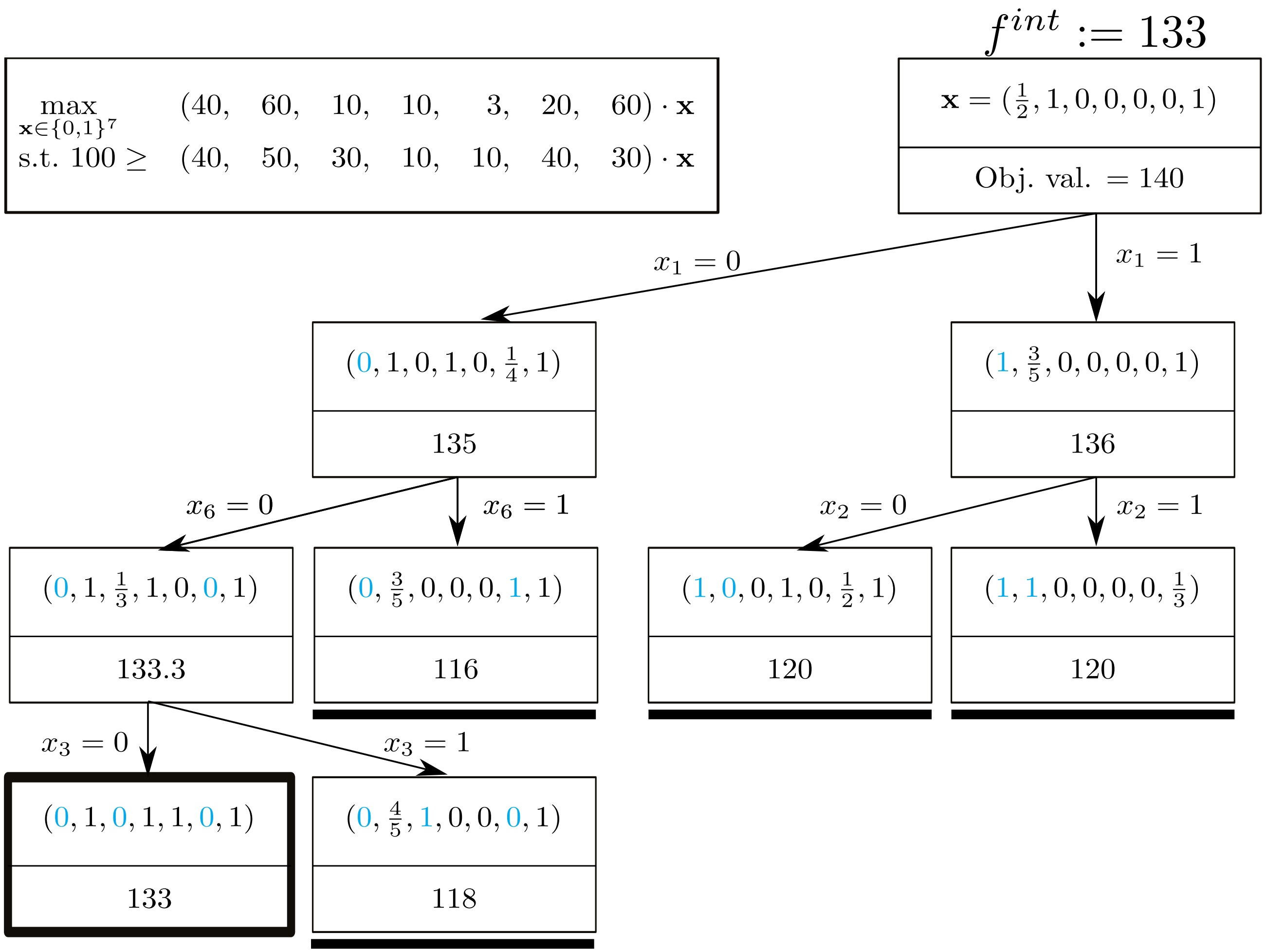
- $f^{LP} < f^{int}$?

  Yes: terminate branch

- Update existing leafs
- Branch

Systematic search: Branch-and-Bound

$$\max_{\mathbf{x}\in\{0,1\}^7} \ (40,\ 60,\ 10,\ 10,\ 3,\ 20,\ 60)\cdot\mathbf{x}$$
$$\text{s.t. } 100 \geq (40,\ 50,\ 30,\ 10,\ 10,\ 40,\ 30)\cdot\mathbf{x}$$



Solve LP relaxation: f^{LP}

- $f^{LP} < f^{int}$?

  Yes: terminate branch

- No leaf to branch

  $\Rightarrow$ Return f^{int}

Systematic search: Branch-and-Bound

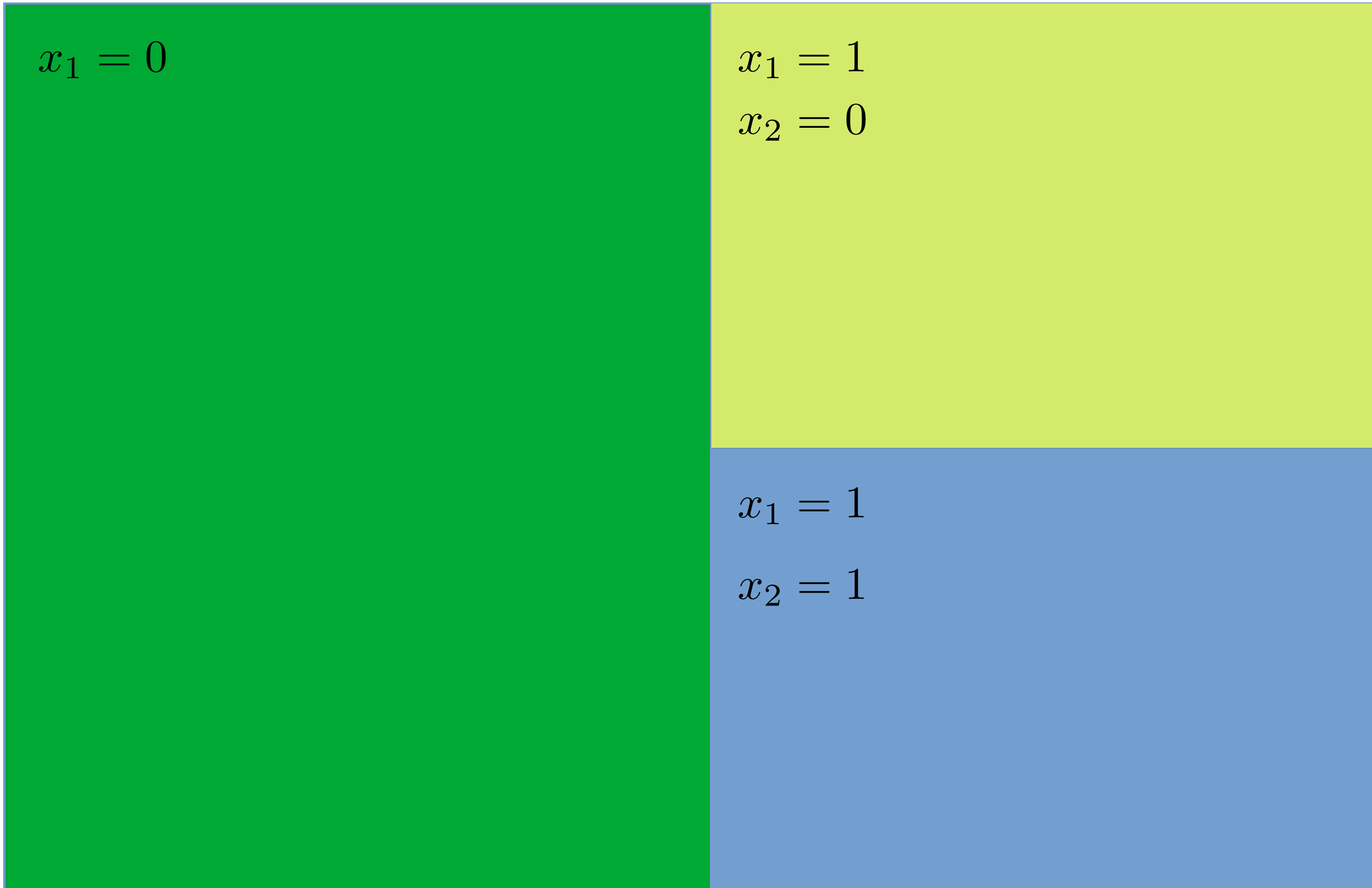
None

Systematic search: Branch-and-Bound

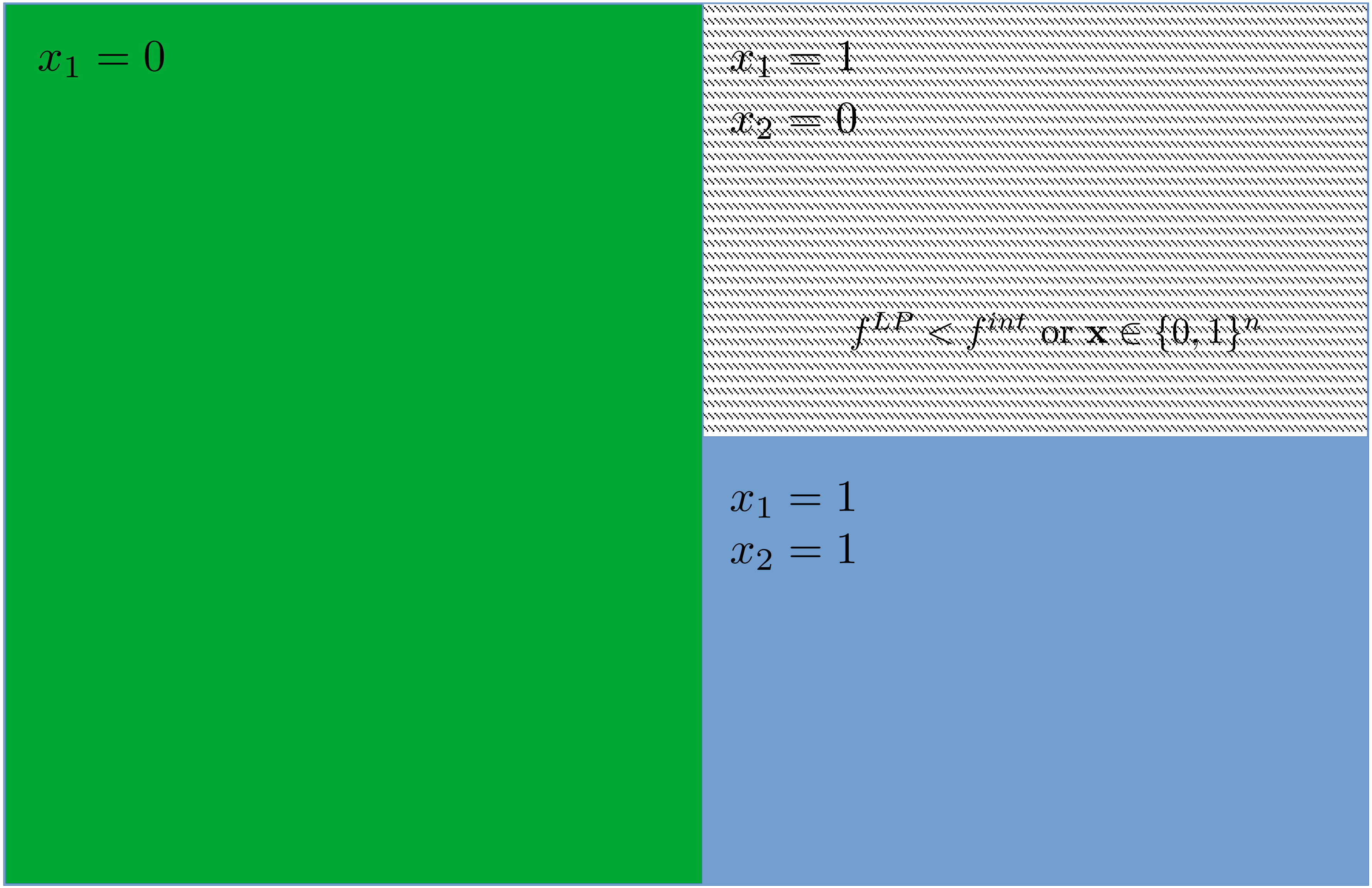
$x_1 = 0$

$x_1 = 1$

Systematic search: Branch-and-Bound



Systematic search: Branch-and-Bound



Systematic search: Branch-and-Bound

$x_1 = 0$

$f^{LP} < f^{int}$ or $\mathbf{x} \in \{0,1\}^n$

$x_1 = 1$
$x_2 = 0$

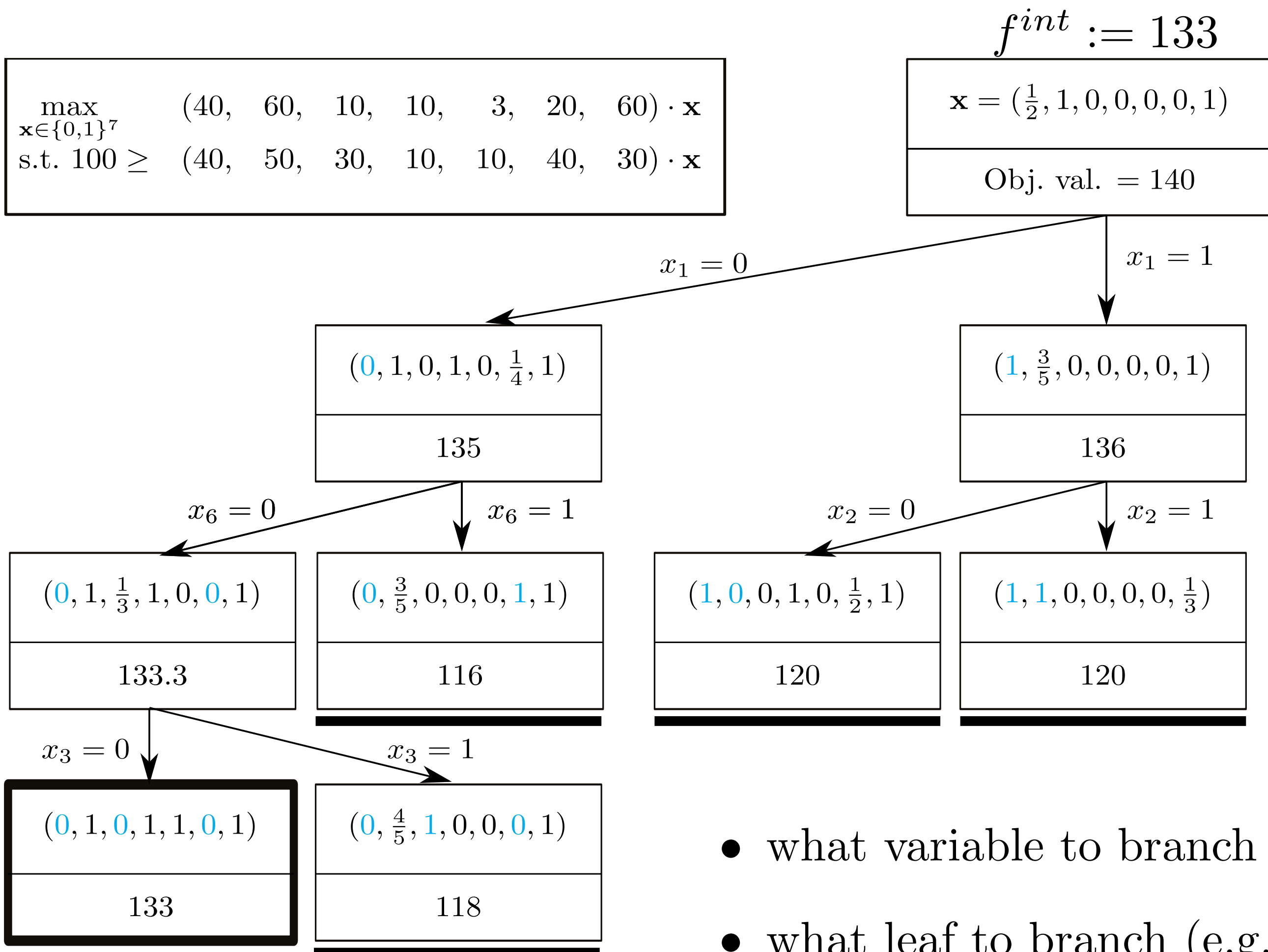
$f^{LP} < f^{int}$ or $\mathbf{x} \in \{0,1\}^n$

$x_1 = 1$
$x_2 = 1$
$x_3 = 0$

$x_1 = 1$
$x_2 = 1$
$x_3 = 1$

Systematic search: Branch-and-Bound

$$\max_{\mathbf{x}\in\{0,1\}^7} \quad (40, \; 60, \; 10, \; 10, \; 3, \; 20, \; 60)\cdot \mathbf{x}$$
$$\text{s.t. } 100 \geq \quad (40, \; 50, \; 30, \; 10, \; 10, \; 40, \; 30)\cdot \mathbf{x}$$



Solve LP relaxation: f^{LP}

- $f^{LP} < f^{int}$?

  Yes: terminate branch

- No leaf to branch

  $\Rightarrow$ Return f^{int}

- what variable to branch on (e.g., closest to 0.5)
- what leaf to branch (e.g., bfs/dfs)
- efficient warm start for branching LPs needed
- additional usage of primal heuristics for better f^{int}

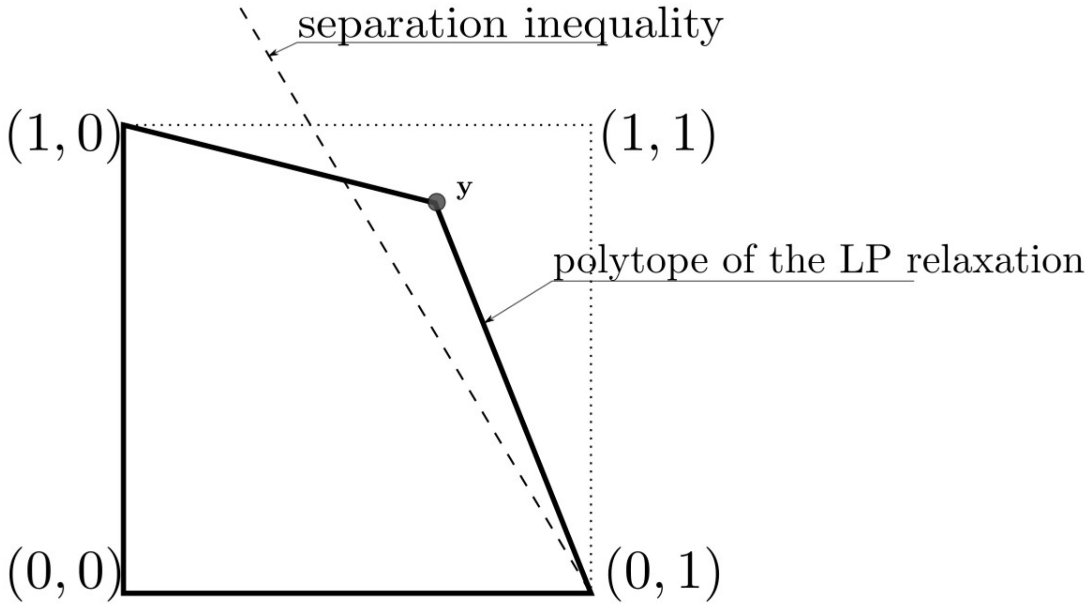
MILP Solvers: **Presolve**, Cutting Planes, Heuristics

$$x_i \in \{0,1\},\ 2x_1 + 2x_2 \leq 1 \quad \Rightarrow \quad 2x_1 + 2x_2 \leq \tfrac{1}{2} \quad \Rightarrow \quad x_1 = x_2 = 0$$

MILP Solvers: **Presolve**, Cutting Planes, Heuristics

$$x_i \in \{0,1\},\ 2x_1 + 2x_2 \leq 1 \quad \Rightarrow \quad 2x_1 + 2x_2 \leq \tfrac{1}{2} \quad \Rightarrow \quad x_1 = x_2 = 0$$

- Fixing variables: $x_i = 0$
- Removing unnecessary constraints
- Identify infeasibilty of constraints: $f^{LP} < f^{int}$
- Logical implications $x_1 = 0 \Rightarrow x_2 = 1$
- Constraints tightening



$$6x_1 + 5x_2 + 7x_3 + 4x_4 + 5x_5 \leq 15$$

LP solution: $x_1 = 0,\ x_2 = 1,\ x_3 = x_4 = x_5 = 3/4$

Note: $7 + 4 + 5 = 16 > 15 \Rightarrow x_3 + x_4 + x_5 \leq 2$

$3/4 + 3/4 + 3/4 = 9/4 > 2 \Rightarrow$ LP solution is cut off

MILP Solvers: Presolve, Cutting Planes, **Heuristics**

$$(\underbrace{x_1, x_2, \ldots,}_{\text{fixed}} x_{i_1}, \underbrace{\ldots,}_{\text{fixed}} x_{i_2}, \underbrace{\ldots,}_{\text{fixed}} x_{i_3}, \underbrace{\ldots, x_n}_{\text{fixed}})$$

Optimize over $x_i,\ i \in I$:

- Random I
- Based on ≥ 2 solutions: $\begin{array}{l}(1, 0, 0, 1)\\(1, 1, 0, 0)\end{array}$

MILP Solvers: Art of problem formulation

- ILP formulation/ LP relaxation
- LP solver selection (primal/dual simplex, interior point)
- Powerful (facet-defining) cutting planes
- Variable pricing (dual to cutting planes)

However...

Outline

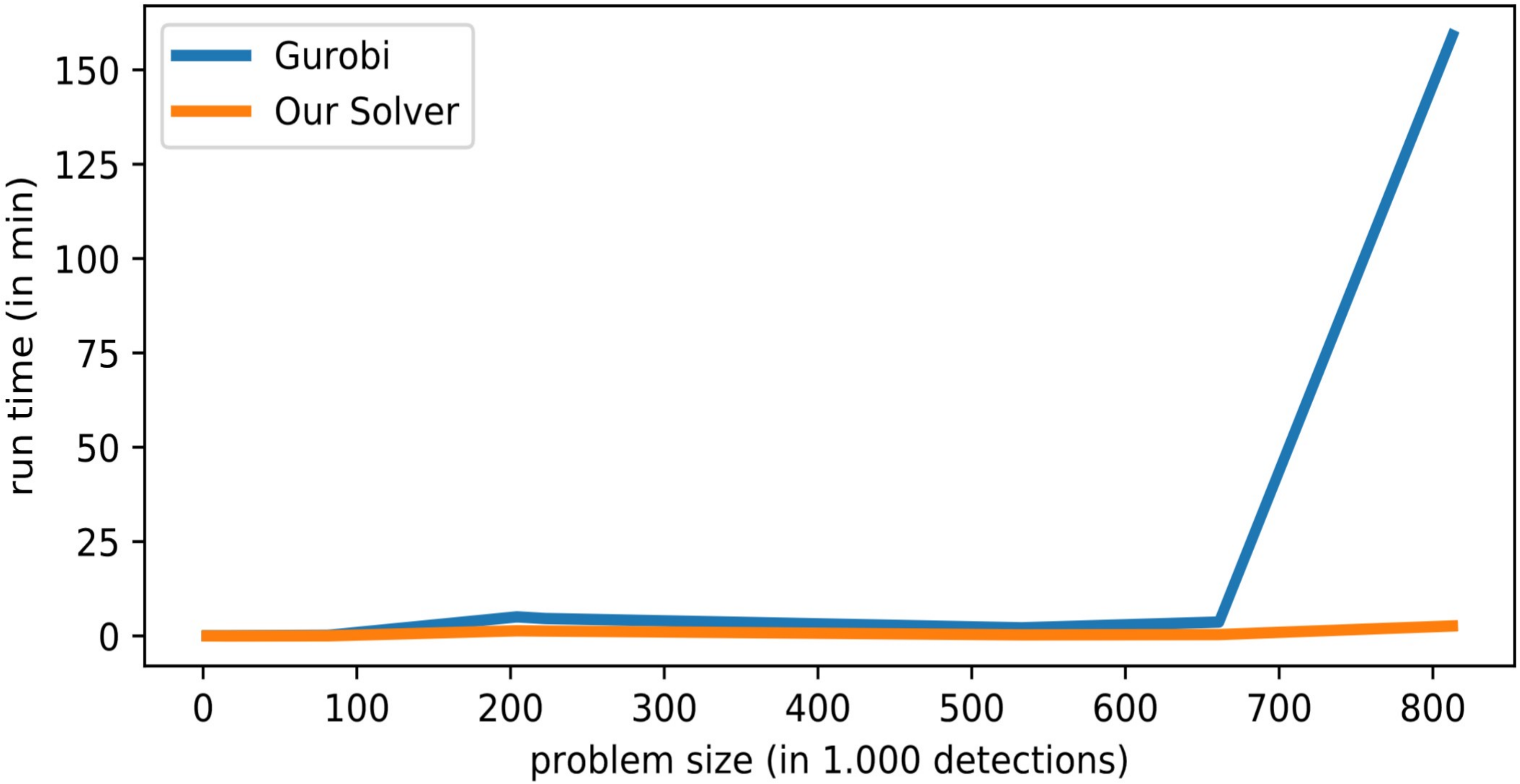
- Integer linear programming:

  A universal language for combinatorial problems

- How off the shelf ILP solvers work

- **What if off the shelf ILP solvers fail?**

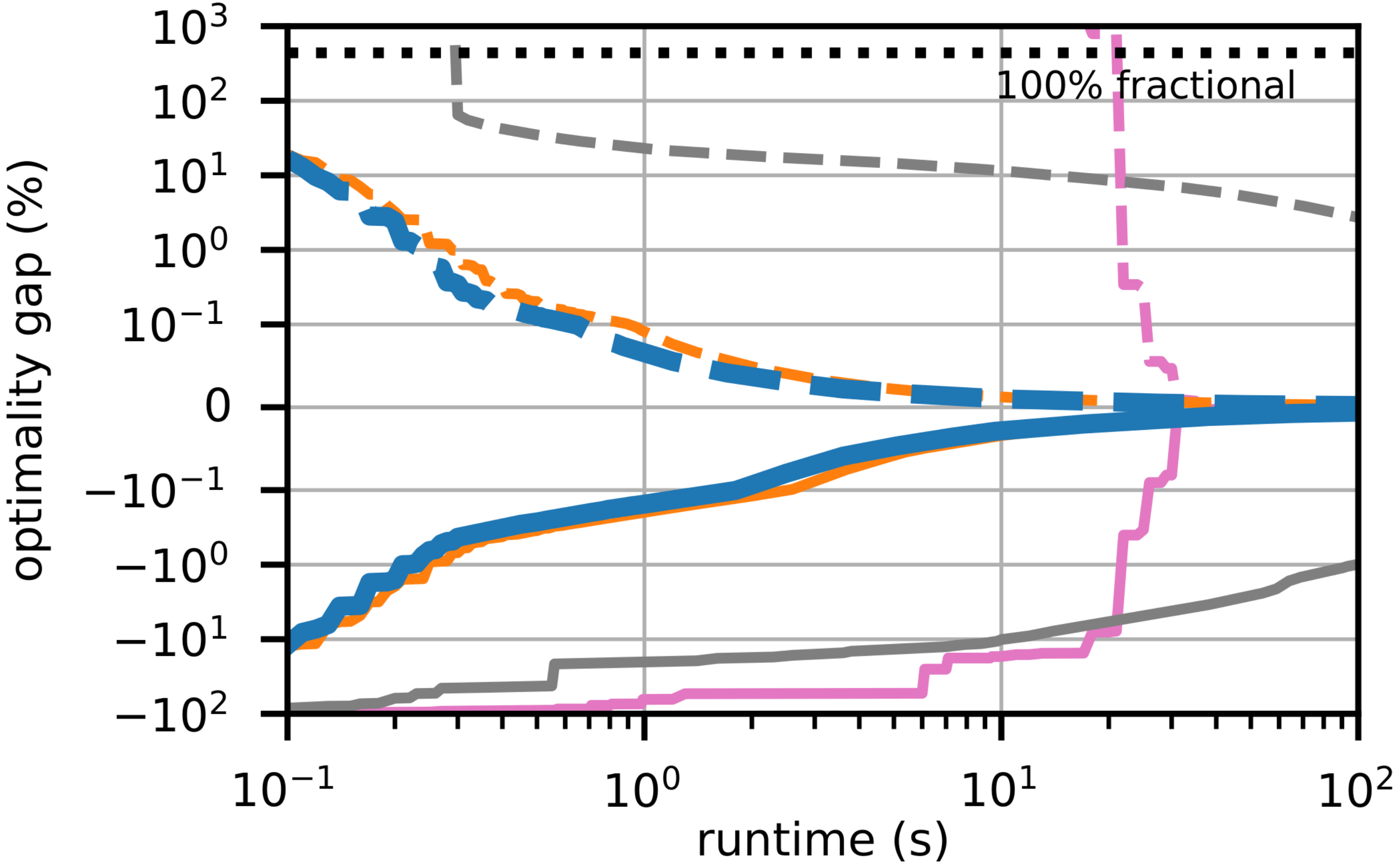
MILP Solver too slow: Reasons



- General LP solver too slow
- Branch-end-bound tree too large:
  - primal heuristic too general/weak

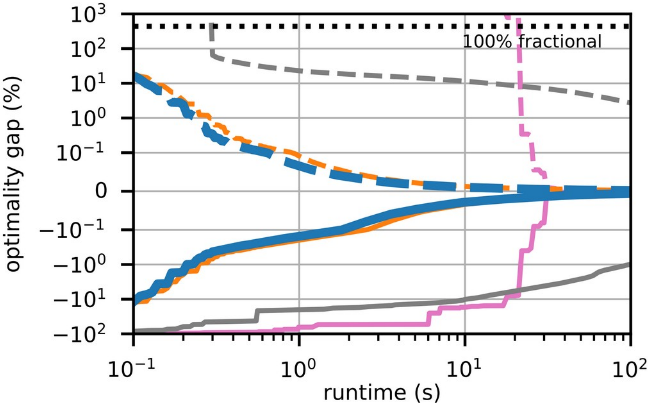
Specialized combinatorial solvers

- First-order non-smooth convex optimization instead of Interior point/simplex :



Specialized combinatorial solvers

- First-order non-smooth convex optimization instead of Interior point/simplex :



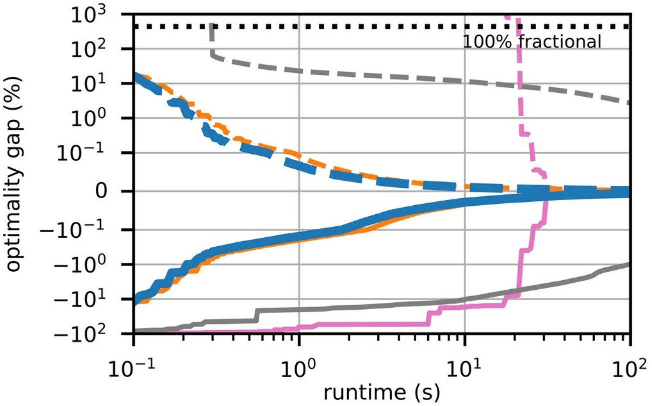
$$\min_{\substack{\mathbf{x}\in[0,1]^n \\ A\mathbf{x}=\mathbf{b}}} \langle \mathbf{c}, \mathbf{x} \rangle = \max_{\boldsymbol{\lambda}} \underbrace{\min_{\mathbf{x}\in[0,1]^n} \left\langle \mathbf{c} + A^\top \boldsymbol{\lambda}, \mathbf{x} \right\rangle - \langle \boldsymbol{\lambda}, \mathbf{b} \rangle}_{\text{concave non-smooth}}$$

$\mathbf{c} + A^\top \boldsymbol{\lambda}^*$ - reduced costs

more informative

Specialized combinatorial solvers

- First-order non-smooth convex optimization instead of Interior point/simplex :



$$\min_{\substack{\mathbf{x}\in[0,1]^n \\ A\mathbf{x}=\mathbf{b}}} \langle \mathbf{c}, \mathbf{x} \rangle = \max_{\boldsymbol{\lambda}} \underbrace{\min_{\mathbf{x}\in[0,1]^n} \left\langle \mathbf{c} + A^\top \boldsymbol{\lambda}, \mathbf{x} \right\rangle - \langle \boldsymbol{\lambda}, \mathbf{b} \rangle}_{\text{concave non-smooth}}$$

$\mathbf{c} + A^\top \boldsymbol{\lambda}^*$ - reduced costs more informative

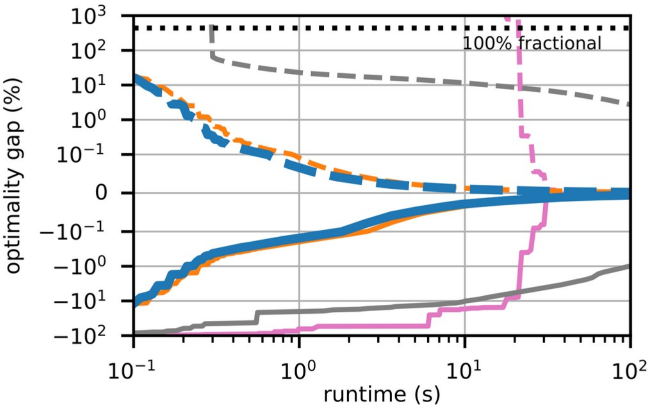
- **Dedicated** primal heuristics:

  - Randomized greedy with reduced costs

$$\mathbf{x}^* \approx \min_{\substack{\mathbf{x}\in\{0,1\}^n \\ A\mathbf{x}=\mathbf{b}}} \left\langle \mathbf{c} + A^\top \boldsymbol{\lambda}, \mathbf{x} \right\rangle$$

Specialized combinatorial solvers

- First-order non-smooth convex optimization instead of Interior point/simplex :



$$\min_{\substack{\mathbf{x}\in[0,1]^n \\ A\mathbf{x}=\mathbf{b}}} \langle \mathbf{c}, \mathbf{x} \rangle = \max_{\boldsymbol{\lambda}} \underbrace{\min_{\mathbf{x}\in[0,1]^n} \left\langle \mathbf{c} + A^\top \boldsymbol{\lambda}, \mathbf{x} \right\rangle - \langle \boldsymbol{\lambda}, \mathbf{b} \rangle}_{\text{concave non-smooth}}$$

$\mathbf{c} + A^\top \boldsymbol{\lambda}^*$ - reduced costs more informative

- **Dedicated** primal heuristics:

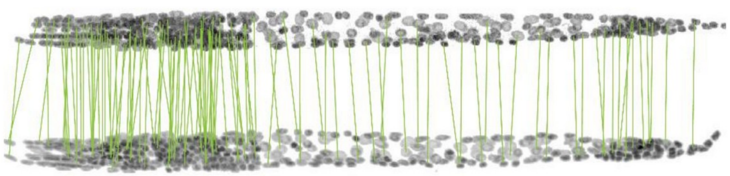
- Randomized greedy with reduced costs

$$\mathbf{x}^* \approx \min_{\substack{\mathbf{x}\in\{0,1\}^n \\ A\mathbf{x}=\mathbf{b}}} \left\langle \mathbf{c} + A^\top \boldsymbol{\lambda}, \mathbf{x} \right\rangle$$

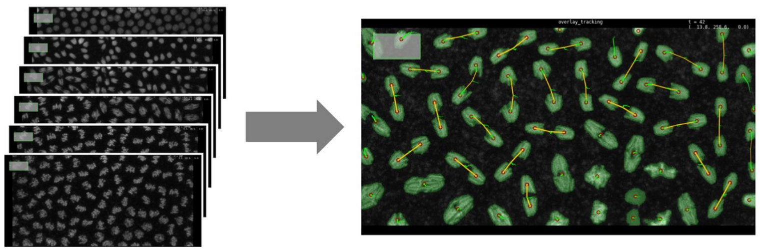
- Local search

$$\left\langle \mathbf{c}, \mathbf{x}^{t+1} \right\rangle \leq \left\langle \mathbf{c}, \mathbf{x}^{t} \right\rangle$$

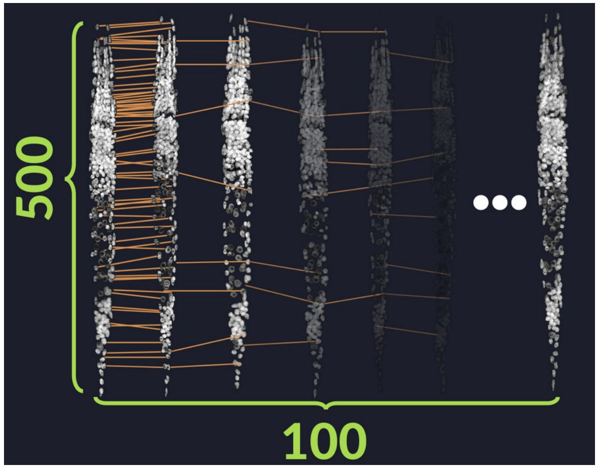
Specialized combinatorial solvers: Applications



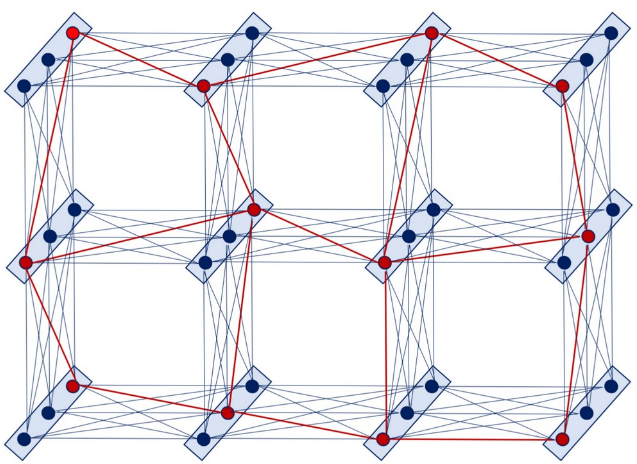
Quadratic assignment



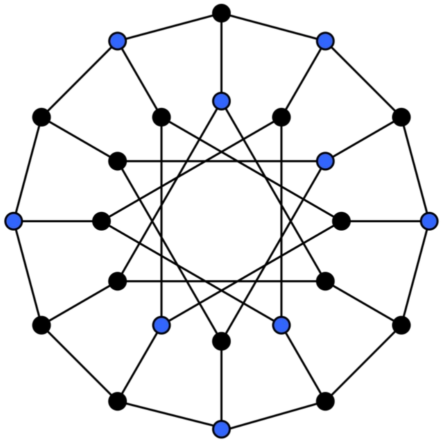
Cell tracking



Multi-graph matching



Discrete labeling



Maximum weight independent set

Want to learn more?

Compact course in WiSe 2025:

Sep. 29 – Oct. 10

More information:

https://hci.iwr.uni-heidelberg.de/content/optimization-machine-learning-WiSe25

Applied Combinatorial Optimization

or Combinatorial Optimization for Artificial Intelligence

Bogdan Savchynskyy
Heidelberg University
bogdan.savchynskyy@iwr.uni-heidelberg.de

June 6, 2025